

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) In a network environment that includes at least a first and second computing system that are capable of communicating messages with each other in a message exchange pattern that defines whether a message is valid or invalid based on a current state of the message exchange pattern, a method for a first computing system enforcing the message exchange pattern by restricting the transmission of invalid messages which do not conform to the current state of the message pattern, to reduce the chance that messages that are invalid at any given point in the message exchange pattern are transmitted to the second computing system to thereby preserve so as to preserve network bandwidth and processing resources, the method comprising the following:

storing a state transition tree for the message exchange pattern in which each node of the tree represents a state of the message exchange pattern and in which a transmission or a receipt of a message causes a state transition to an appropriate next node in the tree;

an act of tracking progress through the message exchange pattern;

at a current node in the state transition tree, detecting a request to transmit a message at the computing system;

an act of detecting a request from a component at the computing system to transmit a message to the second computing system;

based on the current state in the message exchange pattern, determining whether the request to transmit the message renders the message valid or invalid;

when the message is determined to be a valid message given the current node, transmitting the message and transitioning the node to the appropriate next node in the tree;

an act of determining that the message is not a valid message given the tracked progress through the message exchange pattern;

when it is determined that the message is not a valid message to be sent given the current node in the tree, preventing the transmission of the message and an act of notifying the component that the message is not a valid message.

2. (Cancelled)

3. (Original) A method in accordance with Claim 1, wherein the message is a HyperText Transport Protocol (HTTP) message.

4. (Original) A method in accordance with Claim 1, wherein the message is a Simple Object Access Protocol (SOAP) message.

5. (Currently Amended) A method in accordance with Claim 4, wherein a type of message is specified in a SOAP header of the message, wherein determining whether the request to transmit the message renders the message valid or invalid the act of determining that the message is not a valid message given the tracked progress through the message exchange pattern comprises: the following:

~~an act of~~ reading the SOAP header of the message.

6. (Original) A method in accordance with Claim 1, wherein the message is an RMI invocation.

7. (Currently Amended) A method in accordance with Claim 1, wherein the message is a first message, the method further comprising: the following: comprises:

~~an act of~~ detecting a request from a component to transmit a second message to the a second computing system;

~~an act of~~ determining that the second message is a valid message given the tracked progress through the message exchange pattern based on the current state in the message exchange pattern; and

~~an act of~~ transmitting the second message to the second computing system.

8. (Currently Amended) A method in accordance with Claim 7, further comprising:
~~the following:~~

~~an act of identifying a role of the first computing system in the message exchange pattern, wherein the act of determining that the second message is a valid message is performed in light of the identified role.~~

9. (Original) A method in accordance with Claim 7, further comprising the following:

an act of loading state information related to the message exchange pattern from persistent memory to system memory in response to the act of detecting a request to transmit the second message to the second computing system;

an act updating the state information to represent the transmission of the second message to the second computing system upon transmitting the second message to the second computing system; and

saving the updated state information.

10. (Currently Amended) A method in accordance with Claim 1, further comprising:
~~the following:~~

~~an act of loading state information related to the message exchange pattern from persistent memory to system memory in response to the act of detecting a request to transmit the first message to the second computing system.~~

11. (Currently Amended) A method in accordance with Claim 10, further comprising:
~~the following:~~

~~an act of clearing system memory of the updated state information upon the act of notifying the component that the message is not a valid message.~~

12. (Original) A method in accordance with Claim 1, wherein the message exchange pattern includes a plurality of application layer messages.

13. (Original) A method in accordance with Claim 12, wherein the message exchange pattern includes a plurality of protocol layer messages.

14. (Original) A method in accordance with Claim 1, wherein the message exchange pattern includes a plurality of protocol layer messages.

15. (Currently Amended) A method in accordance with Claim 1, wherein the message exchange pattern includes the transmission of one or more messages in which the message exchange pattern is identified and agreed to between the ~~first~~-computing system and ~~the~~ second computing system.

16. (Currently Amended) A method in accordance with Claim 1, further comprising: ~~the following:~~

~~an act of~~ identifying a role of the ~~first~~-computing system in the message exchange pattern, wherein ~~the act of~~ determining that the message is not a valid message is performed in light of the identified role.

17. (Currently Amended) A computer program product for use in a network environment that includes at least a first and second computing system that are capable of communicating messages with each other in a message exchange pattern that defines whether a message is valid or invalid based on a current state of the message pattern, the computer program product for performing a method for a first computing system enforcing the message exchange pattern by restricting the transmission of invalid messages which do not conform to the current state of the message pattern to reduce the chance that messages that are invalid at any given point in the message exchange pattern are transmitted to the second computing system to thereby so as to preserve network bandwidth and processing resources, the computer program product comprising one or more computer-readable storage media having thereon computer-executable instructions comprising that which, when executed by one or more processors of the first computing system, cause the first computing system to perform the following method of claim 1.

an act of tracking progress through the message exchange pattern;

an act of detecting a request from a component to transmit a message to the second computing system;

an act of determining that the message is not a valid message given the tracked progress through the message exchange pattern; and

an act of notifying the component that the message is not a valid message.

18. (Cancelled)

19. (Original) A computer program product in accordance with Claim 18, wherein the one or more computer-readable media includes system memory.

20. (Original) A computer program product in accordance with Claim 18, wherein the one or more computer-readable media includes persistent memory.

21. (Original) A computer program product in accordance with Claim 20, wherein the persistent memory is a magnetic disk.

22. (Currently Amended) A computer program product in accordance with Claim 17, wherein a state transition from a first state to a second state the act of tracking progress through the message exchange pattern comprises; the following:

an act of maintaining a state transition tree representing the message exchange pattern, wherein transitions from a first state to a second state occurs upon the transmission or receipt of one or more valid messages for the first state.

23. (Original) A computer program product in accordance with Claim 17, wherein the message is a HyperText Transport Protocol (HTTP) message.

24. (Original) A computer program product in accordance with Claim 17, wherein the message is a Simple Object Access Protocol (SOAP) message.

25. (Currently Amended) A computer program product in accordance with Claim 24, wherein a type of message is specified in a SOAP header of the message, wherein determining whether the request to transmit the message renders the message valid or invalid the act of determining that the message is not a valid message given the tracked progress through the message exchange pattern comprises; the following:

an act of reading the SOAP header of the message.

26. (Original) A computer program product in accordance with Claim 17, wherein the message is an RMI invocation.

27. (Currently Amended) A computer program product in accordance with Claim 17, wherein the message is a first message, the method further comprising: the following:

an act of detecting a request from a component to transmit a second message to the a second computing system;

an act of determining that the second message is a valid message based on the current state in the message exchange pattern; given the tracked progress through the message exchange pattern; and

an act of transmitting the second message to the second computing system.

28. (Currently Amended) A computer program product in accordance with Claim 27, wherein the one or more computer-readable storage media further have thereon computer-executable instructions that, when executed by the one or more processors, cause the first computing system to perform: the following:

an act of identifying a role of the first computing system in the message exchange pattern, wherein the act of determining that the second message is a valid message is performed in light of the identified role.

29. (Currently Amended) A computer program product in accordance with Claim 27, further comprising: the following:

an act of loading state information related to the message exchange pattern from persistent memory to system memory in response to the act of detecting a request to transmit the second message to the second computing system;

an act of updating the state information to represent the transmission of the second message to the second computing system upon transmitting the second message to the second computing system; and

saving the updated state information.

30. (Original) A computer program product in accordance with Claim 17, further comprising the following:

an act of loading state information related to the message exchange pattern from persistent memory to system memory in response to the act of detecting a request to transmit the first message to the second computing system.

31. (Original) A computer program product in accordance with Claim 30, further comprising the following:

an act of clearing system memory of the updated state information upon the act of notifying the component that the message is not a valid message.

32. (Currently Amended) A computer program product in accordance with Claim 17, wherein the one or more computer-readable storage media further have thereon computer-executable instructions that, when executed by the one or more processors, cause the first computing system to perform: the following:

an act of identifying a role of the first computing system in the message exchange pattern, wherein the act of determining that the message is not a valid message determining whether the request to transmit the message renders the message valid or invalid is performed in light of the identified role.

33 – 36. (Cancelled)